# Chapter 1

1. **Self-Play** Suppose, instead of playing against a random opponent, the reinforcement learning algorithm described above played against itself. What do you think would happen in this case? Would it learn a different way of playing?
    - If a reinforcement learning algorithm only played against random opponents it would probably only learn how to play specifically against those opponents (and not generally). Playing against itself would bypass this restrictions and probably allow it to play more optimally/at a higher level (we see some of this in some of OpenAI's achievements).
2. *Symmetries* Many tic-tac-toe positions appear different but are really the same because of symmetries. How might we amend the reinforcement learning algorithm described above to take advantage of this? In what ways would this improve it? Now think again. Suppose the opponent did not take advantage of symmetries. In that case, should we? Is it true, then, that symmetrically equivalent positions should necessarily have the same value?
    - We can take advantage of the 4-sided symmetry to reduce the state and action space of the environment. This would reduce the memory and computation power since it takes less space to store. If the opponent does not take advantage of symmetries then it would be disadvantageous since if an opponent plays two different moves (one bad and one good, relatively) for two symmetric states the model would value each state the same and not be able to exploit when the opponent plays the bad move.
3. *Greedy Play* Suppose the reinforcement learning player was greedy, that is, it always played the move that brought it to the position that it rated the best. Would it learn to play better, or worse, than a nongreedy player? What problems might occur?
    - It would probably play worse than a nongreedy player (in the long run) since it would not be able to explore effectively. Once it discovers a trajectory that is better than all the others that it knows of, it would continue taking the same path.
4. *Learning from Exploration* Suppose learning updates occurred after all moves, including exploratory moves. If the step-size parameter is appropriately reduced over time, then the state values would converge to a set of probabilities. What are the two sets of probabilities computed when we do, and when we do not, learn from exploratory moves? Assuming that we do continue to make exploratory moves, which set of probabilities might be better to learn? Which would result in more wins?
    - If we do learn from exploratory moves, the distribution of state values will go towards the policy that does the exploring. If we do not learn from exploratory moves, the distribution of state values will approach their true values. If we continue to make exploratory moves it will be better to learn from the non-exploratory moves so we approach the

  true distribution of state values, resulting in winning more.

5. *Other Improvements* Can you think of other ways to improve the reinforcement learning player? Can you think of any better way to solve the tic-tac-toe problem as posed?
   - Since tic-tac-toe is a solved game it would probably be easier to just brute-force all possible move trajectories and choose those that lead to winning more often.